

基本的なアルゴリズムの学習

埼玉県立川越南高等学校, 授業者: 春日井 優

校種: 高等学校, 実施教科: 情報

実施単元: 問題解決とコンピュータの活用, 問題の解決と処理手順の自動化, 実施学年: 3年生

使用したプログラミング言語や実行環境

- Eclipse と Java.
- Eclipse は Pleiades All in One ダウンロードサイト (<http://mergedoc.sourceforge.jp>) からダウンロードしたものを使用 (授業時は Eclipse4.2)。

教室の設備・環境

- **実施場所:** コンピュータ教室
 - コンピュータ教室にはクライアント機として, 教員機 1 台と生徒機 42 台のデスクトップコンピュータが整備。
 - Windows7, CPU:Core i5, HDD:500GB, メモリ:4GB。21.5 インチのワイドスクリーンのディスプレイが接続。
 - サーバ機が 1 台整備されており, ユーザ認証およびファイルサーバとして利用。
 - 有線 LAN にサーバ機とクライアント機が接続されており, 教材等を配布可能。
 - 中間モニタが 2 人に 1 台の割合で整備されており, 教員機・生徒機の画面転送が可能。
 - 実物投影機が整備されており, コンピュータの画面だけでなく, プリントなどの印刷物等の提示も可能。
 - Eclipse は教員機と生徒機すべてのコンピュータにインストールし, デスクトップ上のショートカットから起動可能。

授業の様子

```
double temperature[] = { 26.0, 22.7,
int days = 3;
double average;
average = temperature[0]+temperature
int days = 3;
double average;
average = temperature[0] + temperature[
System.out.println(average);
}
```

✖ aberage を変数に解決できません
5 個のクイック・フィックスが使用可能です:
○ ローカル変数 'average' を作成します
□ フィールド 'average' を作成します
◆ 'average' に変更します
○ パラメーター 'average' を作成します
✖ 代入を削除します

✖ 構文エラーがあります。"}" を挿入して ClassBody を完了してください
フォーカスするには 'F2' を押下

Eclipse 上でプログラムを作成する。変数名の間違いや括弧が正しく対応していないなどの文法的な誤りを指摘し修正候補を示すなどの支援が即座に受けられる。



プログラムを作成するのに苦労している生徒が, 早くプログラムができた生徒の説明を聞ける機会を作る。

プログラムと実行結果が同じウインドウ内に表示されることにより, 正しい動作かどうかのフィードバックを受けやすい。

```
package science;
public class MaxTemperature {
public static void main(String[] args) {
// TODO 自動生成されたメソッド・スタブ
double temperature[] = { 26.0, 22.7, 27.3 };
double max;
double win;
if (temperature[0] > temperature[1]) {
win = temperature[0];
} else {
win = temperature[1];
}
if (win > temperature[2]) {
max = win;
} else {
max = temperature[2];
}
System.out.println("最高気温は" + max + "°Cです。");
}
```

<終了> MaxTemperature [Java アプリケーション] C:\pleiades\java\bin\javaw.exe (2014/11/26 18:25:32)
最高気温は27.3°Cです。

全体指導計画（全体 8 時間計画）

1. アルゴリズムとフローチャート……………(0.5 時間)
2. 基本的なアルゴリズム
 - (ア) …… 順次構造・選択構造・繰り返し構造……………(1.5 時間)
 - (イ) …… 簡単なアルゴリズム【本時 1/2】……………(2 時間)
3. プログラムの活用
 - (ア) …… 探索……………(2 時間)
 - (イ) …… 並べ替え……………(2 時間)

授業の概要

授業の目標：基本的なアルゴリズムを理解し、簡単な処理を自動化することができる（知識・理解）（技能）。また、同じ処理のアルゴリズムを比較することにより、処理手順を評価できる（思考・判断・表現）。

評価規準

- A・・・基本的なアルゴリズムを、さまざまな処理手順により自動化することができる。データの個数が増えた場合にも、少ない変更により処理できるプログラムを記述できる。
- B・・・基本的なアルゴリズムを、Java でプログラムを記述することにより、処理を自動化することができる。
- C と判断した生徒への手立て・・・コンピュータを使わないで処理する方法を、手作業により行わせる。その手順を 1 つ 1 つプログラム言語に対応させて記述させる。

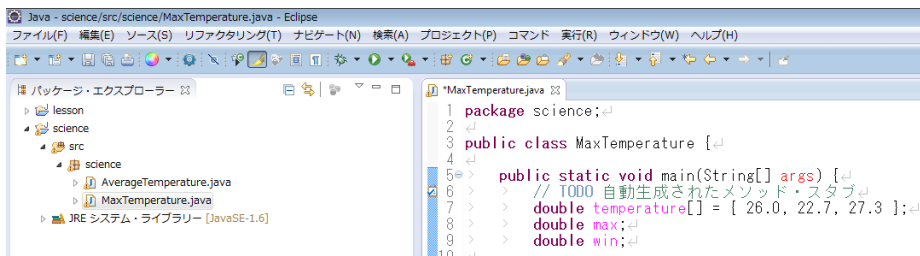
指導過程：（2. 基本的なアルゴリズム (イ) 簡単なアルゴリズム【本時 1/2】）

	教員の働きかけ	予想される生徒の反応	指導上の留意点
導入 (5 分)	1.前時までに学習した順次構造・選択構造・繰り返し構造について確認する。	1.if や for などの命令の使い方や書式を確認する。	
<div style="border: 1px solid black; padding: 5px; background-color: #e0f0ff;"> <p>【導入場面のここがポイント】</p> <ul style="list-style-type: none"> ・構造の概念的な説明だけでなく、どのような記述をすればどのように動かすかを意識させる。 </div>			
	2.本時の課題を提示する。	2.本時の課題を知る。	
<div style="border: 1px solid black; padding: 5px; background-color: #e0f0ff;"> <p>平均や最大値などを求める基本的なプログラムをつくろう。</p> </div>			
展開 (40 分)	3.日常的に行っている操作と対応付けて手順を考えさせる。	3.平均や最大などを求めるプログラムを考える。 ◎単純な方法により、Java のプログラムを記述することができる。 ◎データが増えた場合にも大幅な変更をしないで動作するプログラムが考えられる。	・評価の場面 【知識・理解】【技能】 ・生徒の作成したプログラム
	4.完成した生徒のプログラムを提示し、全体で発表させる。	4.他の生徒が工夫した点や生徒間による手順の違いなど気づいたことをワークシートにまとめさせる。	・評価の場面 【思考・判断・表現】 ・ワークシート
<div style="border: 1px solid black; padding: 5px; background-color: #e0f0ff;"> <p>【展開場面のここがポイント】</p> <ul style="list-style-type: none"> ・プログラムをどのように書いたら良かがわからない生徒は、近くの生徒と話したりネットで検索したりするが、その点はあまり制限せず、他の生徒やネット上の情報からのヒントをもとに理解を深め、プログラムを記述できるようにする。 ・「プログラムの 1 つ 1 つの命令がどのように働いているか」を理解しているか確認することが必要である。 ・正しい結果が出ることにのみ着目している生徒がみられるが、手順をひとつひとつ追って処理を記述していくプロセスの方が重要であることを伝える。 </div>			
まとめ (5 分)	5.本時の授業を通して、学んだことや手順の工夫の仕方についてワークシートにまとめさせる。	5.本時の授業を通して学んだことや手順の工夫の仕方について考える。 ◎日常的に無意識に行っていることも、コンピュータで実行するには手順をしっかりと考えることが必要である。 ◎同じことを実行するにもさまざまな手順があり、多くのデータを扱って処理できるようにするためには工夫が必要である。	5.データが増えた場合にも、プログラムの変更をしないで処理できるようにするにはどのようなことが必要かを考えるように促す。 ・評価の場面 【思考・判断・表現】 ・ワークシート
<div style="border: 1px solid black; padding: 5px; background-color: #e0f0ff;"> <p>【まとめ場面のここがポイント】</p> <ul style="list-style-type: none"> ・データの値や個数などが変化した場合に対応できるように、具体的な処理からより一般化させるための工夫について考えさせる。 </div>			

生徒の作品例

どのようなプログラムか

- 配列内に過去 3 日分の最高気温が与えられている。
- 順次構造・選択構造・繰り返し構造を用いて、3日間の最高気温を求める（必ずしもすべての構造を用いる必要はない）。
- 求めた3日間の最高気温を出力する。
- 配列のデータを増やすことにより、日数を増やすことができる（ただし、正しく動作させるためには、下記のプログラムでは一部を書き換える必要がある）。



```

1 package science;
2
3 public class MaxTemperature {
4
5     public static void main(String[] args) {
6         // TODO 自動生成されたメソッド・スタブ
7         double temperature[] = { 26.0, 22.7, 27.3 };
8         double max;
9         double win;
10    }

```

Eclipse の実行ボタンを押すと、プログラムが実行され実行結果の最高気温が表示される。

```

<終了> MaxTemperature [Java アプリケーション] C:\pleiade
最高気温は27.3°Cです。

```

どのように動くか

- ① 3日間の日ごとの最高気温のデータと3日間の最高気温を代入するための変数を宣言している。この部分は教員から配布したものである。
- ② 1日目と2日目の気温を比較して、気温が高い方を勝者とし、変数 win に代入している。
- ③ ②の勝者 win と3日目の気温を比較して、気温が高い方を3日間の最高気温とし、max に代入している。
- ④ 最高気温を出力している。この部分は、①と同様に教員から配布したものである。

```

1 package science;
2
3 public class MaxTemperature {
4
5     public static void main(String[] args) {
6         // TODO 自動生成されたメソッド・スタブ
7         double temperature[] = { 26.0, 22.7, 27.3 };
8         double max;
9         double win;
10
11     ② if (temperature[0] > temperature[1]) {
12         win = temperature[0];
13     } else {
14         win = temperature[1];
15     }
16
17     ③ if (win > temperature[2]) {
18         max = win;
19     } else {
20         max = temperature[2];
21     }
22
23     ④ System.out.println("最高気温は" + max + "°Cです。");
24 }
25
26 }

```

別の生徒のプログラム

- ⑤ 最高気温の初期値として、最高気温になりえない値(-9999)を代入している。
- ⑥ 1日目から日数の分だけ、前日までの最高気温(max)とその日の最高気温との比較を繰り返す。その日の最高気温の方が大きいならば、前日までの最高気温を保持する変数 max に、その日の最高気温を代入している。
- ⑦ 最終的に代入された max の値が3日間の最高気温となっている。

```

1 package science;
2
3 public class MaxTemperature {
4
5     public static void main(String[] args) {
6         // TODO 自動生成されたメソッド・スタブ
7         double temperature[] = { 26.0, 22.7, 27.3, 28.4 };
8         double max;
9         int days = 4;
10
11     ⑤ max = -9999;
12     for (int i = 0; i < days; i++) {
13
14         ⑥ if (temperature[i] > max) {
15             max = temperature[i];
16         }
17     }
18
19     ⑦ System.out.println("最高気温は" + max + "°Cです。");
20 }
21
22 }

```

生徒の変容や生徒の感想より

生徒の変容

- 初めてプログラミングを行う生徒がほとんどのため、開始当初にはコンピュータの処理方法に戸惑いを感じる生徒が多くいたが、処理方法を理解した後は自分たちでプログラムを考える姿勢が見られた。
- すべて教えてもらえるという姿勢の生徒が見られたが、動作するプログラムをつくった生徒が現れ始めると自分で考える雰囲気になった。
- 生徒が書いたプログラムによる処理が実際に動作する経験を通して、専門化されていたコンピュータの処理に興味や関心を持つようになった。

生徒の感想

- はじめはプログラムが動かなくて嫌になっていたけれど、友人や先生に教えてもらってプログラムが動いたときはとても嬉しかった。
- コンピュータはさまざまなことができ便利だけれど、そのためのプログラムをつくることは大変だと思った。
- もっとプログラミングを勉強して、コンピュータでいろんなことができるようになりたいと思った。

この授業のお勧めポイントと実施上の留意点

題材のおすすめポイントと留意点

- 長いプログラムでの 10 行程度の題材を扱うので、プログラムの構造が比較的単純である。それにも関わらず、さまざまな場面で利用できる題材である。
- 教科書では順次構造・選択構造・繰り返し構造を行った後にすぐに探索や並べ替えのアルゴリズムが掲載されていることが多いが、この題材を扱うことで内容の飛躍が少なくなる。
- 簡単なプログラムの処理手順を考えさせることで、論理的思考力を養うことができる。
- コンピュータは簡単に高機能な処理を行うと思っている生徒が多いため、基本的なアルゴリズムを考える際には、制約となる事項（二数の比較しかできないなど）を明確にすることが重要である。

指導方法のおすすめポイントと留意点

- 実際に生徒にプログラムを書かせることにより、コンピュータが処理を行うしくみについて経験を通して理解させることができる。
- Eclipse を利用することにより、プログラミング言語の文法に則っていない箇所やスペルミスの発見が容易になり、40 人の生徒を対象にした授業でも、生徒がプログラミングを行う際にそれぞれの生徒への支援となる。
- 教科書に掲載されているプログラムは Visual Basic や JavaScript が多い。言語が異なることにより、教材研究の際に不安を感じることもあるかもしれないが、プログラム言語が異なっても、同じアルゴリズムで処理できることを示す教材にもなる。

その他

- 基本的なアルゴリズムを理解することが目標である場合には、大抵のプログラミング言語は順次構造・選択構造・繰り返し構造を記述できるため、どのプログラミング言語を選択するかはあまり重要ではない。生徒がプログラムを記述した際に、エラーや実行結果を生徒自身にフィードバックできる環境を整える方が重要である。

参考

- 中山清喬・国本大悟：スッキリわかる Java 入門，インプレス(2011)